Atwood Machine Solver

Creating a CASIO-Basic program to solve simple pulley problems

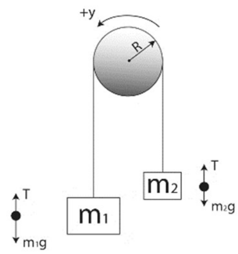
Introduction

Physics problems at the high-school level are definitely thought-provoking, but they are not all unique. There are categories and types of problems that are often repeated, even on the same test. While the situation and numbers may change, much of the derivations stay the same and rarely require more than two equations. Pulley problems, also called Atwood machines, are a good example of these problems. The scenarios and numbers may change but the method used to arrive at a solution remains the same. Thus, these problems turn into a sort of muscle memory and students do not think much when solving. “Plug and chug” is a common term used to describe this phenomenon. However, this can lead to numerous calculator errors such as multiplying the wrong numbers or using the wrong constant. The researcher ran into these problems often and it hindered his grade greatly.

In order to avoid these errors, the researcher sought to automate the process. It was convenient that graphing calculators came with an option to write programs which could be executed on the calculator itself. Using this, the researcher believed that automating this process was possible and he would never have to make a simple mistake on an Atwood machine again.

Review of Literature

Rev. George Atwood was a tutor at Trinity College, Cambridge in the late 1700s. However, his name is now attached to a machine which provides for an infamous set of problems that help demonstrate the tension force. Atwood machines usually consist of two masses connected by a wire or string which hangs over a pulley. There are many alterations to the problem, some of which can be very tedious because of all the forces involved. These problems accurately help to demonstrate how the tension force behaves and how the gravitational force can affect a system.



A typical Atwood Machine problem

In many physics classrooms, to solve these problems, a graphing calculator is preferred over a regular four-function calculator. This is due mostly to the many features that come with the calculator. One of the popularized brands for these calculators is Casio. Casio is credited for releasing the world's first entirely electric compact calculator and the company continues to develop their calculators. Like many graphing calculators, they have their own programming language, called CASIO-Basic, that the calculator can understand and compile. CASIO-Basic is not a widely-used programming language because it is limited to being used only on Casio's graphing calculators. Even if the language could be used on other calculators, it would still be limited because of the lack of functionality it has. The language was developed to increase the uses of a graphing calculator because time can be saved when the calculations are automated through programs.

However, the language itself is very difficult to understand. At first, the researcher believed that the program had to be typed on the calculator itself and that would have been a pain because of the time it would have needed to type and access all the necessary functions. However, after researching, he found that Casio provided a linking software online that also had a way to type the programs on a PC and then transfer it to the calculator to compile. This way, the only process that was time-consuming was the transfer of the program to the calculator to run. However, this was greatly preferred over typing it directly on the calculator. The software, known as CASIO FA-124, also improves the readability of the language.

In addition to the lack of properly developed methods of writing in CASIO-Basic, the language itself is not properly developed. Many of the methods it provides come with perplexing syntax. The syntax does not seem to be well-thought out. While there is a wide range of available methods, the syntax seems to change for everything. The use of labels instead of proper loops makes the program hard to edit because of the fear that the logic of the program to be altered. That is just one example of many the researcher faced.

Success Statement

The Researcher is going to use CASIO-Basic to build a program that will automate the process of solving simple Atwood machine problems.

Materials

- Casio *fx-9750GII* graphing calculator
- A computer with Microsoft Windows OS and CASIO FA-124

Procedure

1. Type the following code into the CASIO FA-124 program editor

```
Filename:AW
'Filename:SCIFAIR↵
Lbl 0↵
Menu "which type","regular",1,"modified",2,"inclined",3,"exit",4↵
'↵
'↵
'↵
'Regular Atwood Machine↵
Lbl 1↵
"M#E5D1"?→A↵
"M#E5D2"?→B↵
(9.8×(B-A))÷(A+B)→S↵
↵
(A×9.8)+(A×S)→T↵
(B×9.8)-T→F↵
"Tension: "↵
T◢
"Net Force: "↵
F◢
"Acceleration: "↵
S◢
Do↵
LpWhile Getkey≠31↵
ClrText↵
Goto 0↵
'↵
'↵
'↵
'Modified Atwood Machine↵
Lbl 2↵
Menu "Friction?","Yes",5,"No",6↵
'Friction↵
Lbl 5↵
"M#E5D1"?→A↵
"M#E5D2"?→B↵
"#E64Bs"?→M↵
"#E64Bk"?→U↵
If (B×9.8)>(M×A×9.8)↵
Then (B×9.8)-(U×A×9.8)→F↵
F÷(A+B)→S↵
```

```
(B×9.8)-(B×S)→T↵
Else 0→F↵
F÷(A+B)→S↵
(B×9.8)+(B×S)→T↵
IfEnd↵
"Tension: "↵
T◢
"Net Force: "↵
F◢
"Acceleration: "↵
S◢
Do↵
LpWhile Getkey≠31↵
ClrText↵
Goto 0↵
'No Friction↵
Lbl 6↵
"M#E5D1"?→A↵
"M#E5D2"?→B↵
(B×9.8)÷(A+B)→S↵
B×(9.8-S)→T↵
B×9.8→F↵
"Tension: "↵
T◢
"Net Force: "↵
F◢
"Acceleration: "↵
S◢
Do↵
LpWhile Getkey≠31↵
ClrText↵
Goto 0↵
'↵
'↵
'↵
'Inclined Plane↵
Lbl 3↵
Menu "Friction?","Yes",7,"No",8↵
```

```
'Friction↵
Lbl 7↵
"M#E5D1"?→A↵
"M#E5D2"?→B↵
"#E64Bs"?→M↵
"#E64Bk"?→U↵
"#E647"?→Tθptch↵
If B>A↵
Then If (M×cos Tθptch×A×9.8)>(B×9.8)↵
Then 0→F↵
F÷(A+B)→S↵
(B×9.8)+(B×S)→T↵
Else (B×9.8)-(U×cos Tθptch×A×9.8)→F↵
F÷(A+B)→S↵
(B×9.8)-(B×S)→T↵
IfEnd↵
Else If ((M×A×9.8×cos Tθptch)+(B×9.8))>(A×9.8×sin Tθptch)↵
Then (0→F)↵
F÷(A+B)→S↵
A×(S+(9.8×sin Tθptch)-(M×9.8×cos Tθptch))→T↵
```

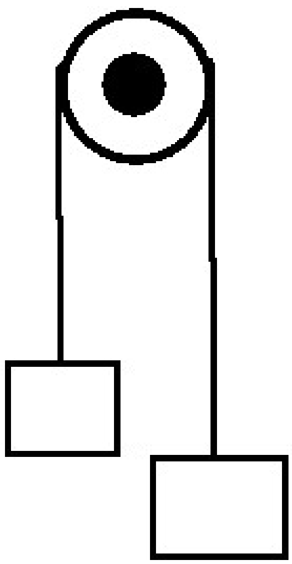
```
Else If (A×9.8×sin Tθptch)>((M×A×9.8×cos Tθptch)+(B×9.8))↵
Then (A×9.8×sin Tθptch)-((U×A×9.8×cos Tθptch)+(B×9.8))→F↵
F÷(A+B)→S↵
(A×9.8)+(A×S)→T↵
Else (((U×A×9.8×cos Tθptch)+(B×9.8))-(A×9.8×sin Tθptch))→F↵
F÷(A+B)→S↵
A×(S+(9.8×sin Tθptch)-(M×9.8×cos Tθptch))→T↵
IfEnd↵
IfEnd↵
"Tension: "↵
T◢
"Net Force: "↵
F◢
"Acceleration: "↵
S◢
Do↵
LpWhile Getkey≠31↵
ClrText↵
```

```
Goto 0↵
'No Friction↵
Lbl 8↵
"M#E5D1"?→A↵
"M#E5D2"?→B↵
"#E647"?→Tθptch↵
If B>A↵
Then ((B×9.8)-(sin Tθptch×9.8×A))→F↵
F÷(A+B)→S↵
↵
(B×9.8)-(B×S)→T↵
Else ((sin Tθptch×9.8×A)-(B×9.8))→F↵
F÷(A+B)→S↵
(B×9.8)+(B×S)→T↵
IfEnd↵
"Tension: "↵
T◢
"Net Force: "↵
F◢
"Acceleration: "↵
S◢
Do↵
LpWhile Getkey≠31↵
ClrText↵
Goto 0↵
Lbl 4↵
Stop↵
```

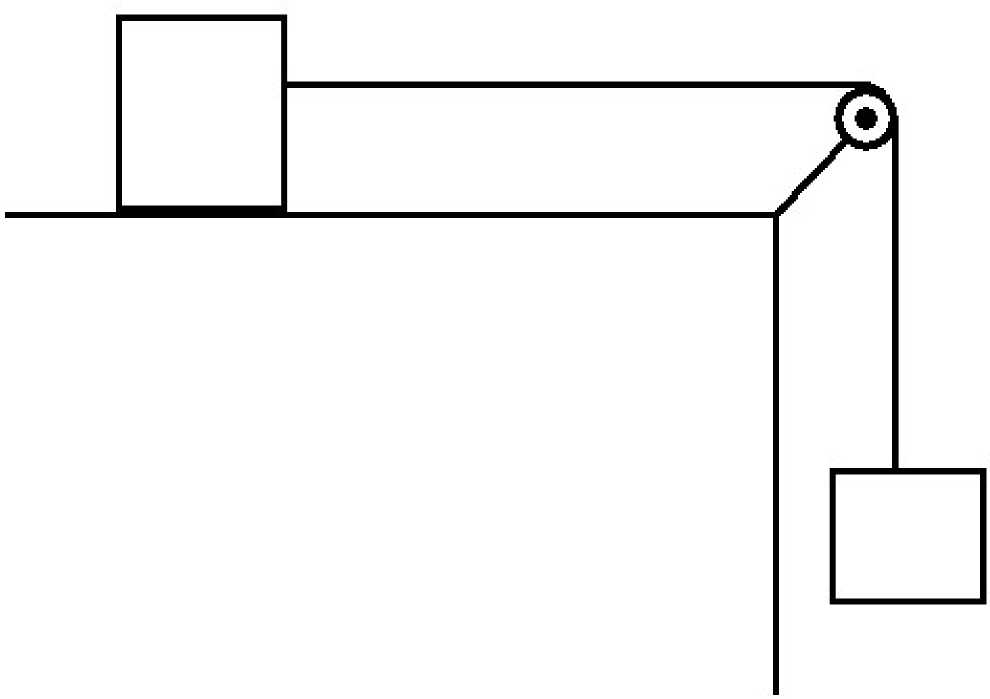
2. Connect your Casio calculator via USB
3. Transfer the program to your Casio calculator
4. Go to the program menu and execute the program

Data

Regular Atwood Machine

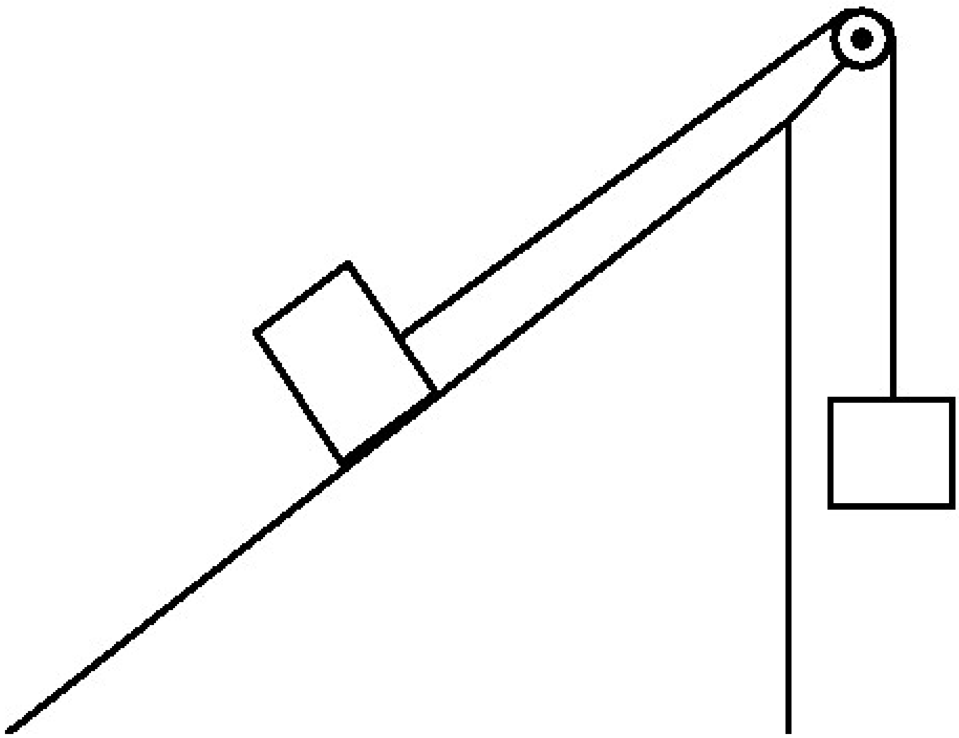


Modified Atwood Machine



Yash Gharat
4B
Dr. Raj

Inclined Atwood Machine



Yash Gharat
4B
Dr. Raj

Results obtained through derived calculations

Regular Atwood Machine

Mass 1	Mass 2	Net Force	Acceleration	Tension
1	2	9.8	3.27	13.07
3	4	9.8	1.40	33.60
10	14	39.2	1.63	114.33
9	12	29.4	1.40	100.80

Modified Atwood Machine (No friction)

Mass 1	Mass 2	Net Force	Acceleration	Tension
2	1	9.8	3.27	6.53
3	4	39.2	5.60	16.80
14	10	98	4.08	57.17
9	9	88.2	4.90	44.10

Modified Atwood Machine (Friction)

Mass 1	Mass 2	μs	μk	Net Force	Acceleration	Tension
2	1	0.8	0.6	0.00	0.00	-9.80
3	4	0.25	0.2	31.85	4.76	20.16
14	10	0.9	0.85	0.00	0.00	-98.00
9	12	0.1	0.05	108.78	5.39	52.92

Inclined Atwood Machine (No friction)

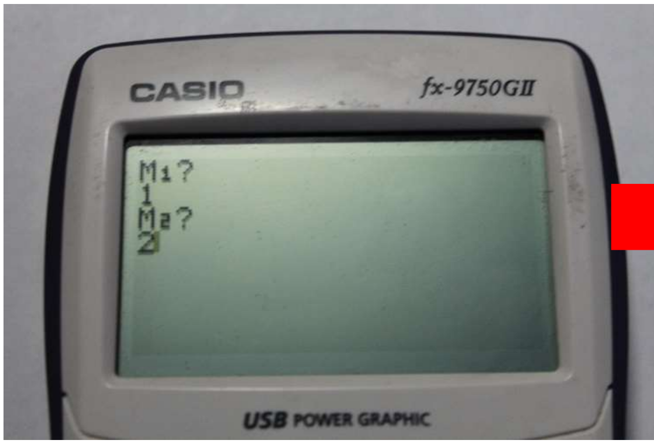
Mass 1	Mass 2	θ	Net Force	Acceleration	Tension
2	1	25	1.52	0.51	9.29
3	4	15	31.59	4.51	21.15
14	10	30	-29.40	-1.23	85.75
9	12	25	80.33	3.83	71.70

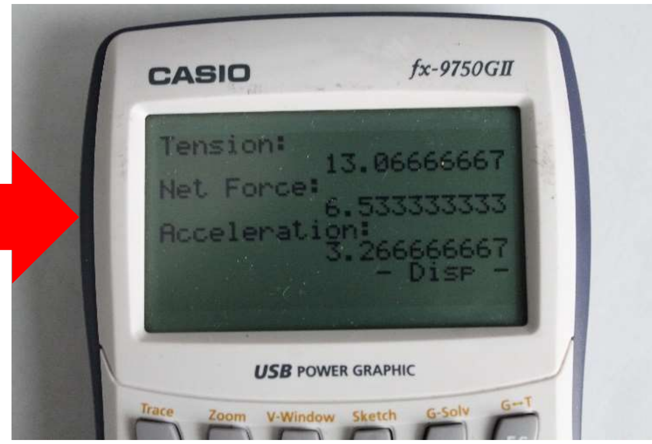
Inclined Atwood Machine (Friction)

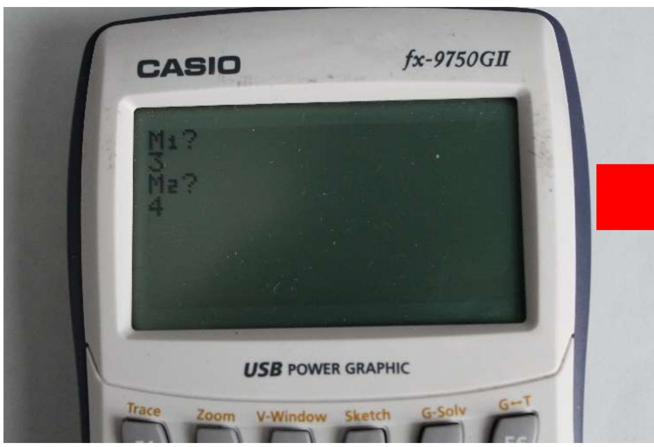
Mass 1	Mass 2	θ	μs	μk	Net Force	Acceleration	Tension
1	2	25	0.8	0.6	14.27	4.76	10.09
3	4	15	0.25	0.2	33.52	4.79	20.05
10	14	30	0.9	0.85	65.06	2.71	99.25
9	12	25	0.1	0.05	113.60	5.41	52.68

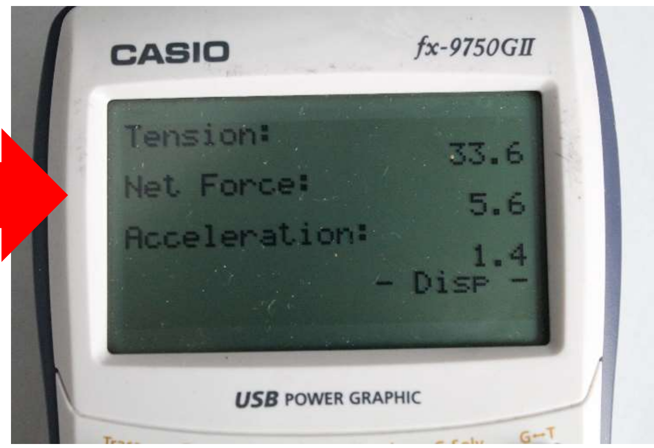
Results obtained through executing program

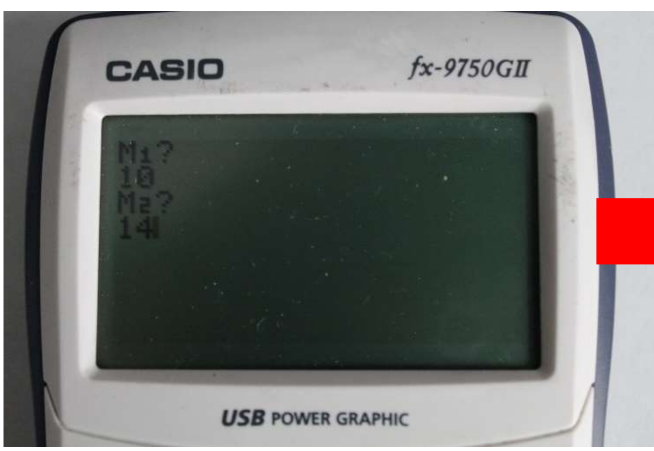
Regular

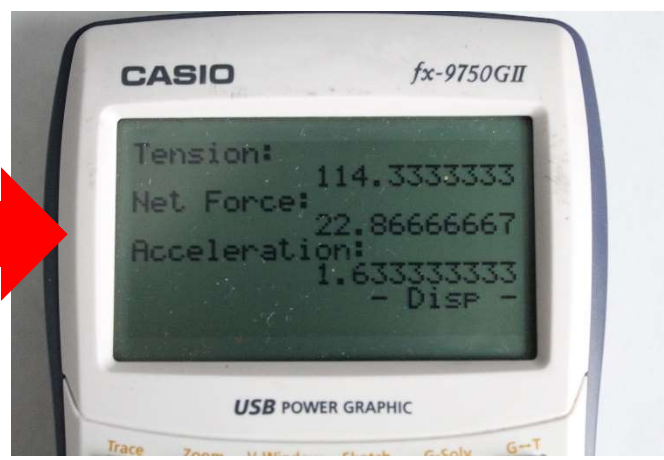


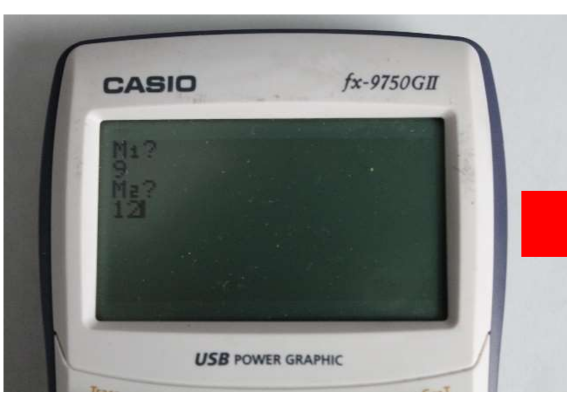


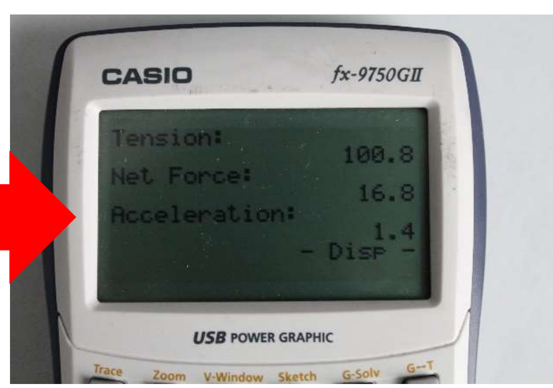






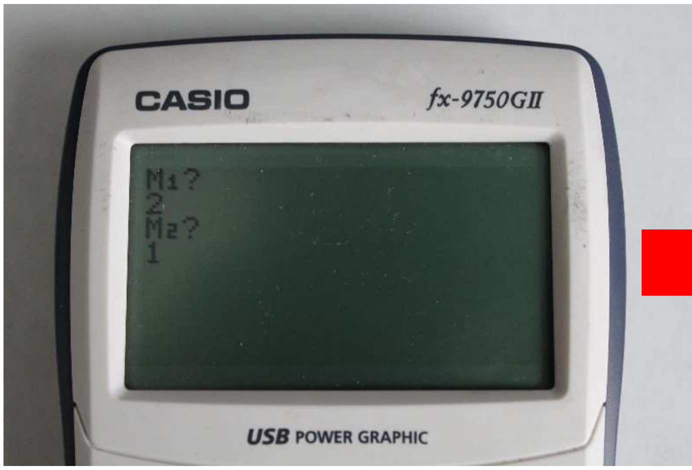


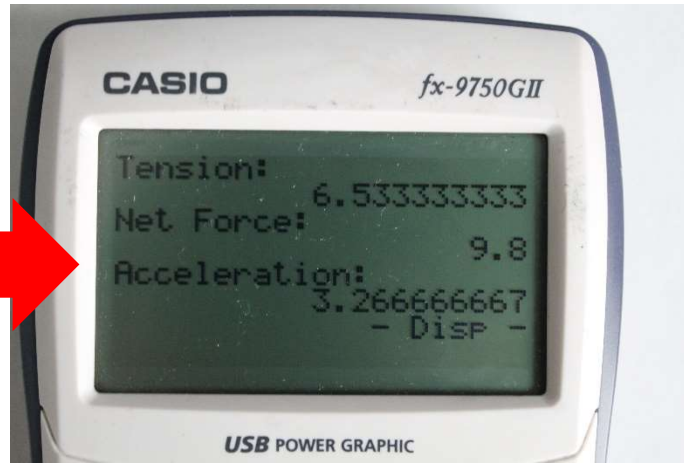


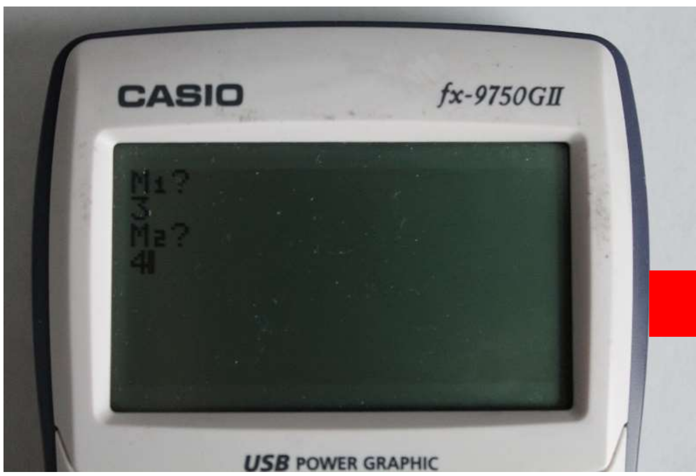


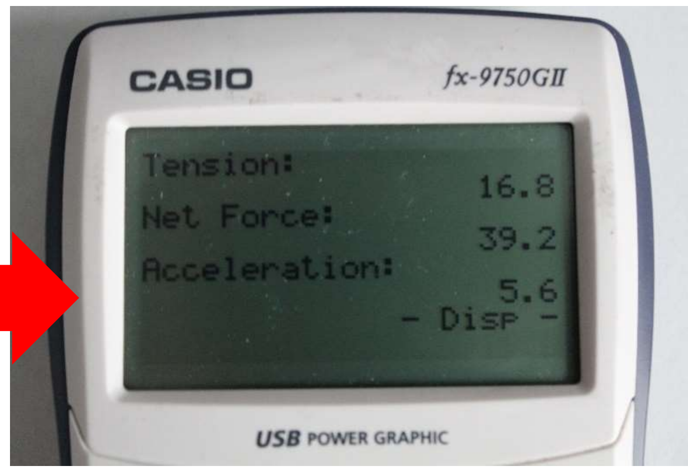
Yash Gharat
4B
Dr. Raj

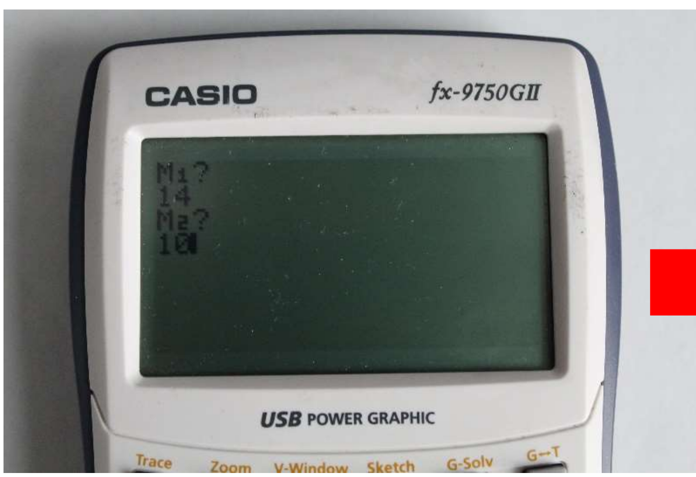
Modified (No Friction)

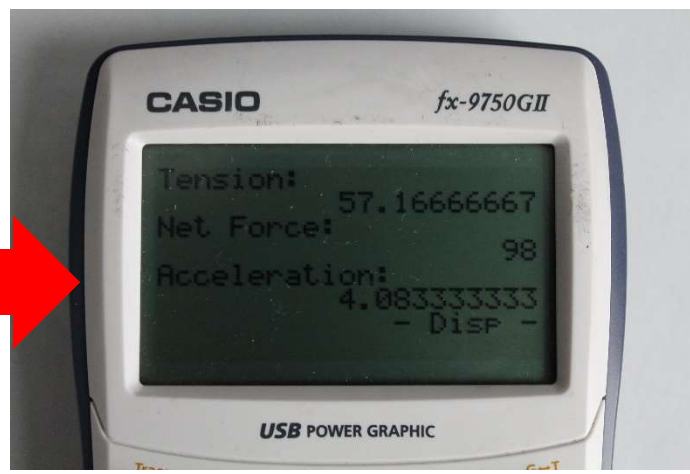


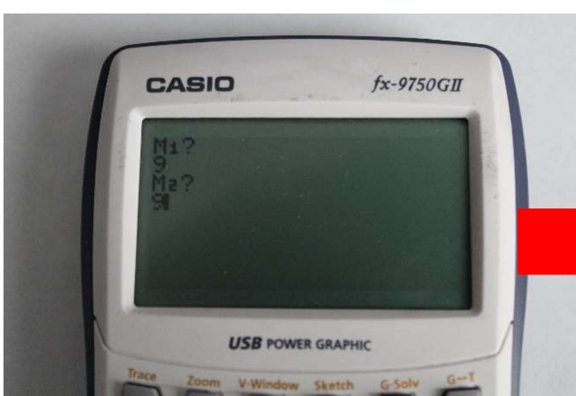


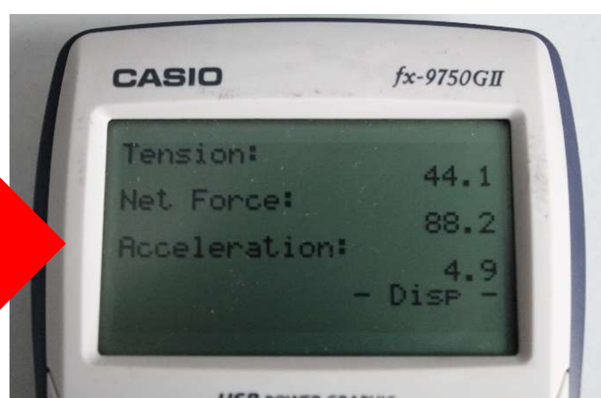






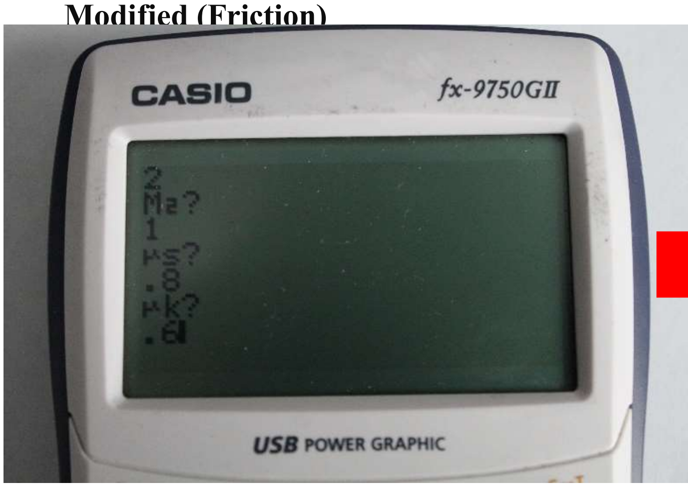


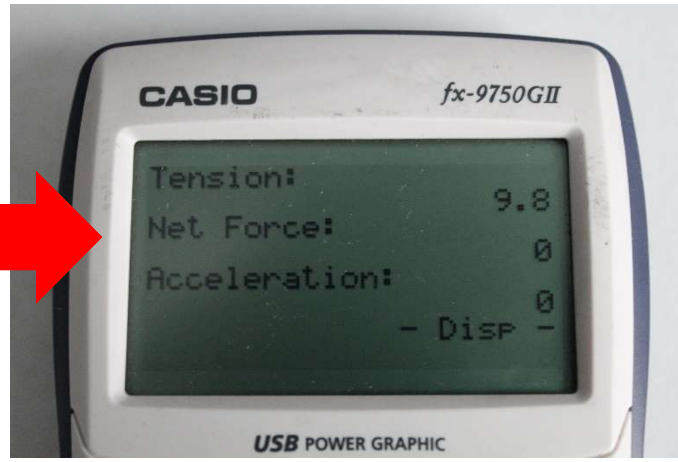


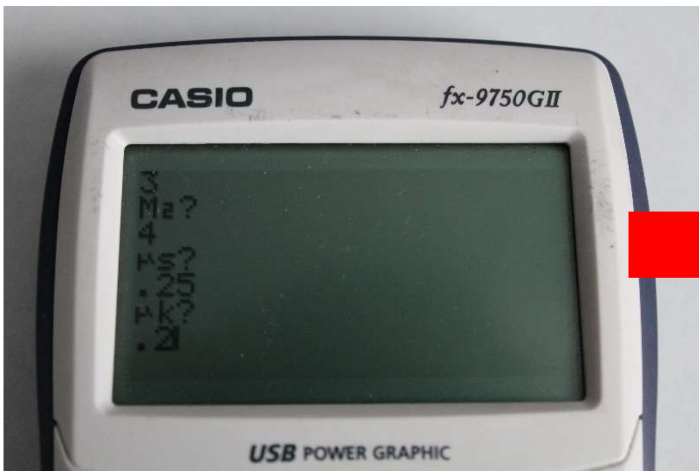


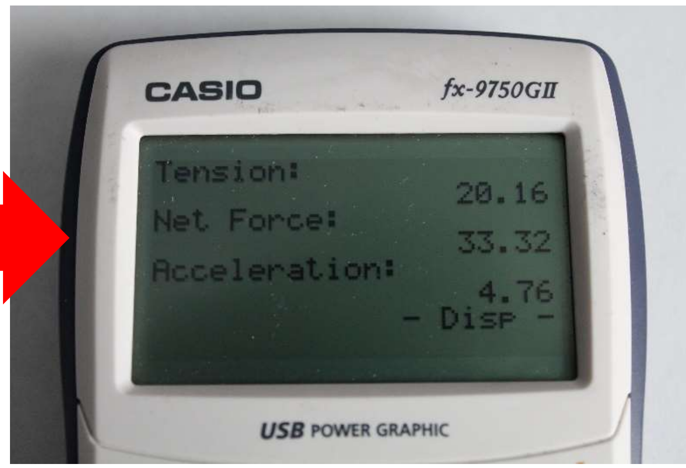
Yash Gharat
4B
Dr. Raj

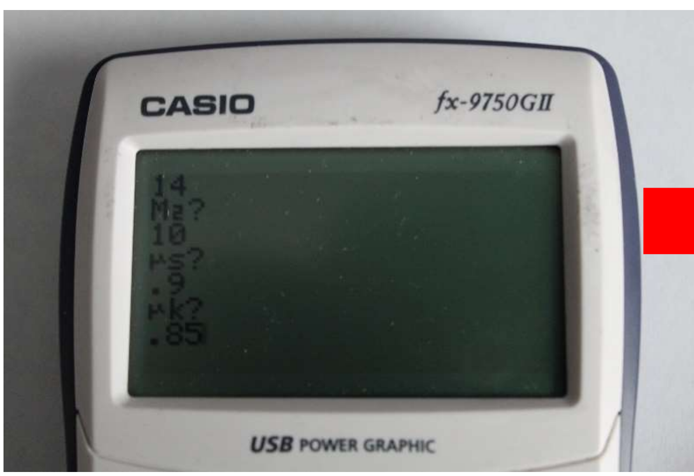
Modified (Friction)

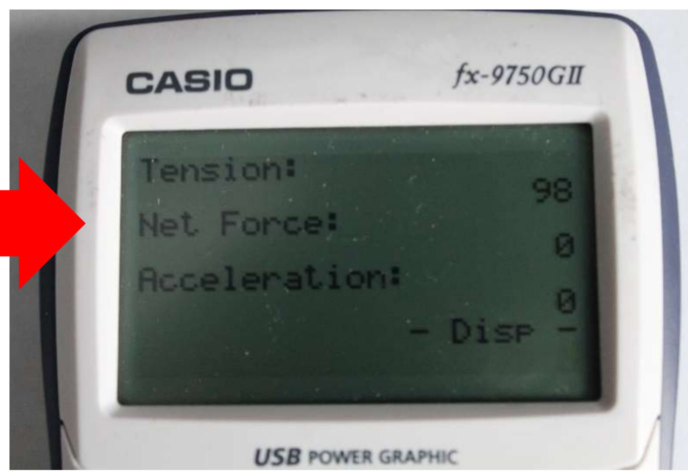


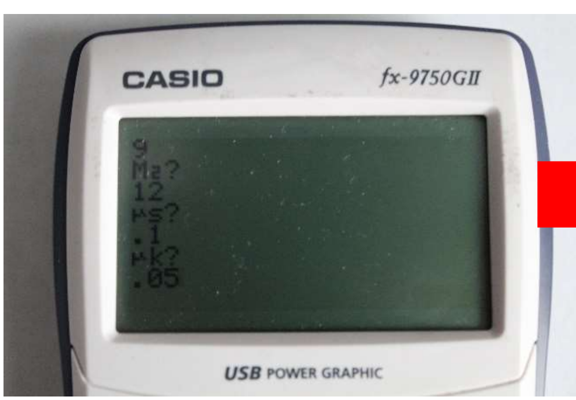


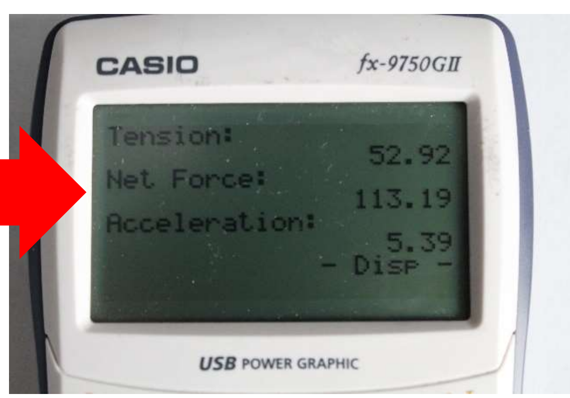






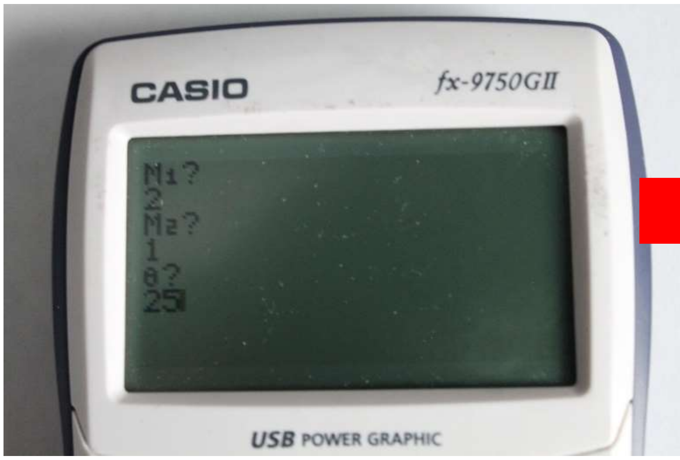


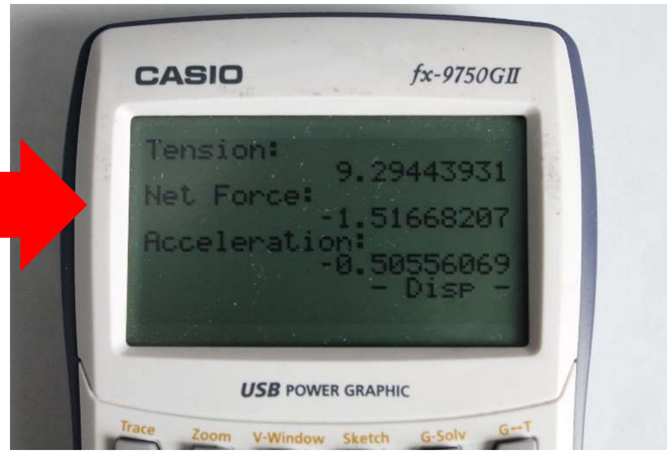


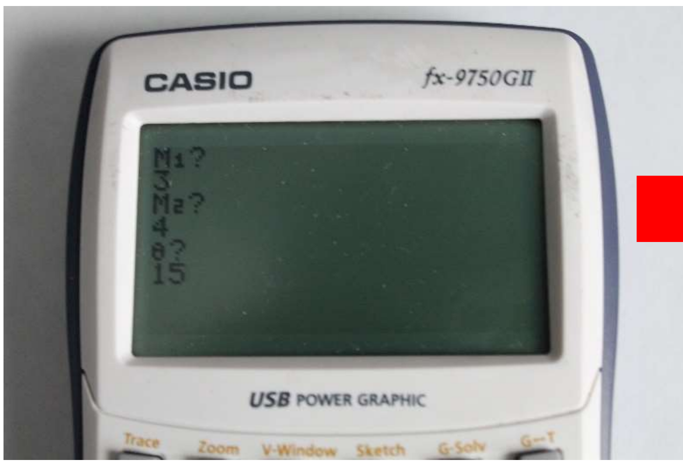


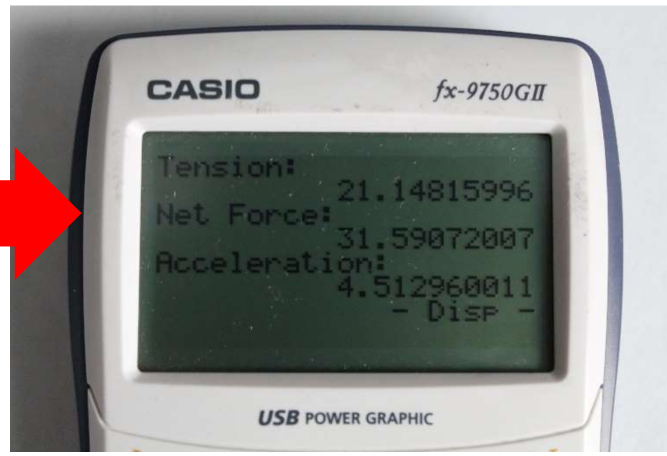
Yash Gharat
4B
Dr. Raj

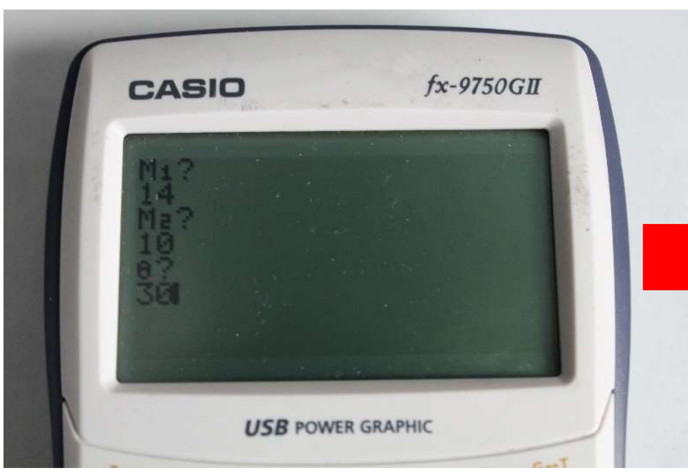
Inclined (No Friction)

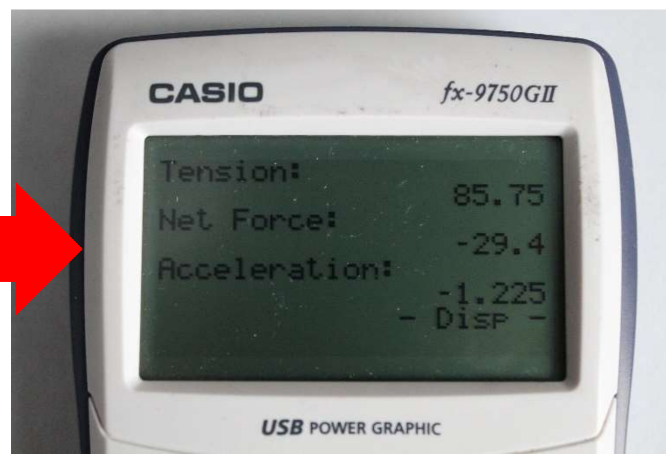


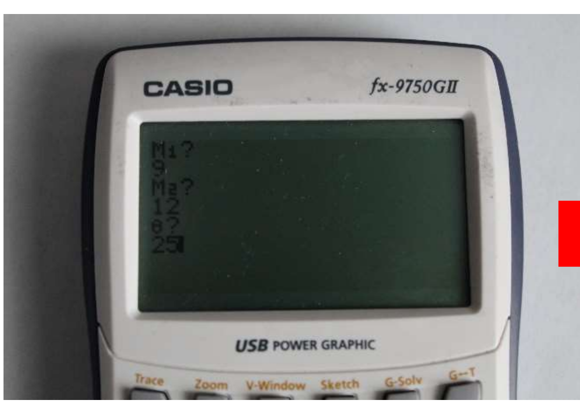


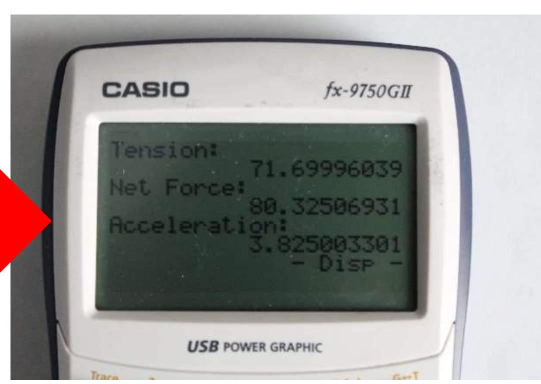






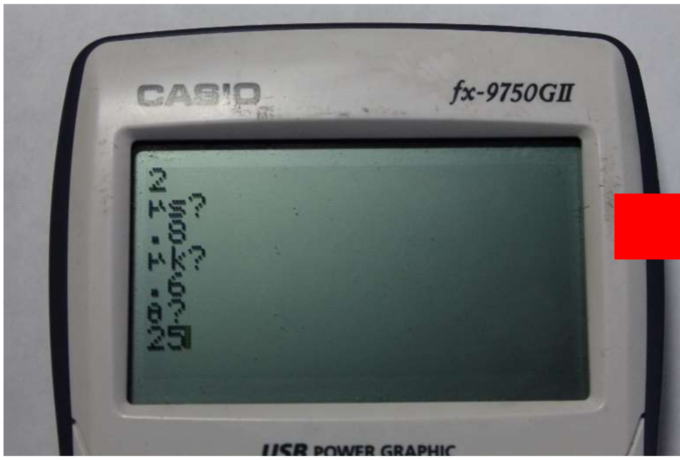


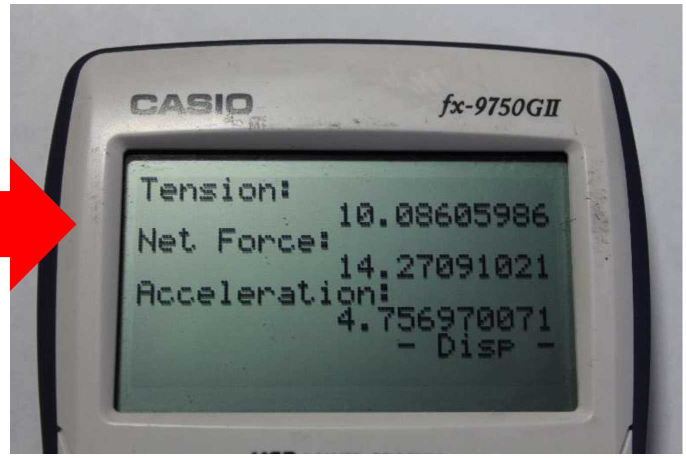


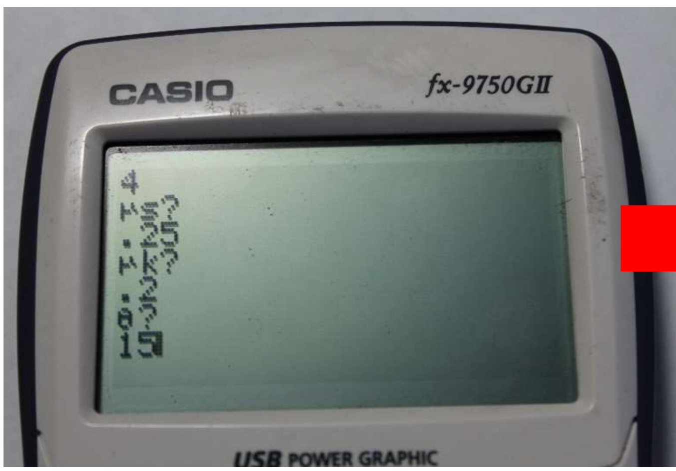


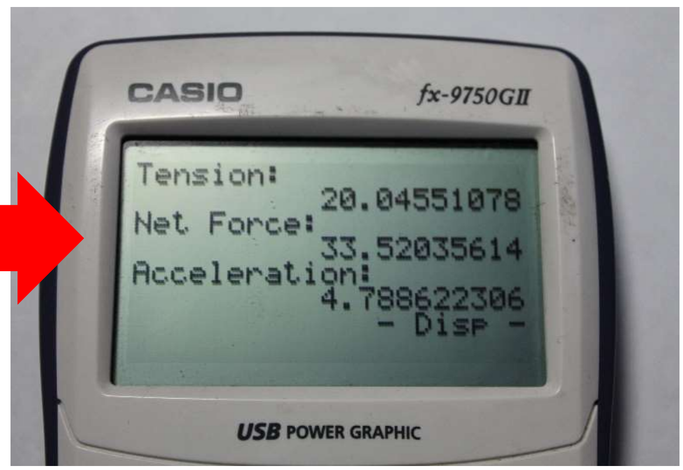
Yash Gharat
4B
Dr. Raj

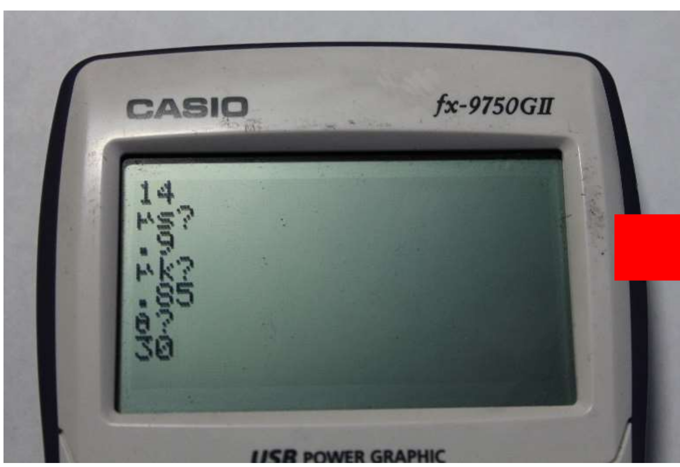
Inclined (Friction)

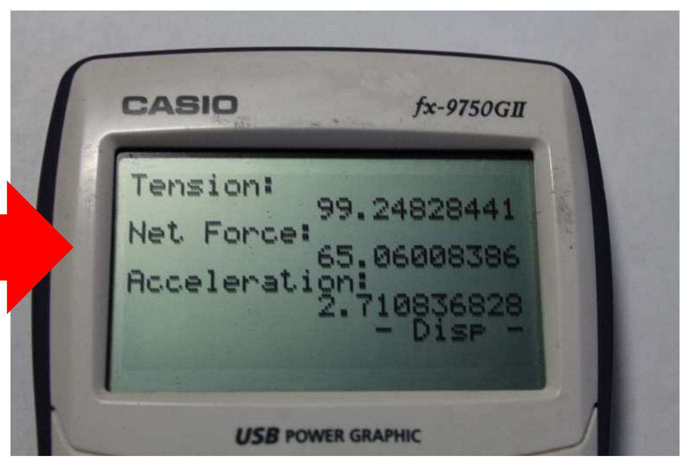


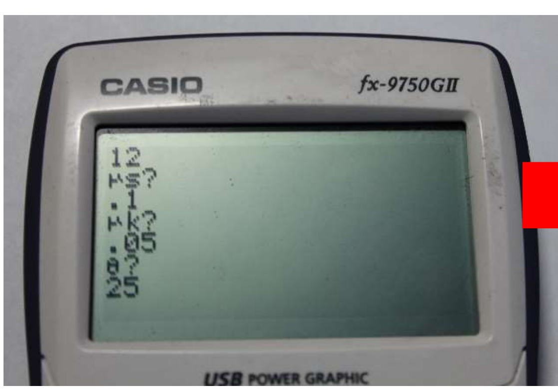


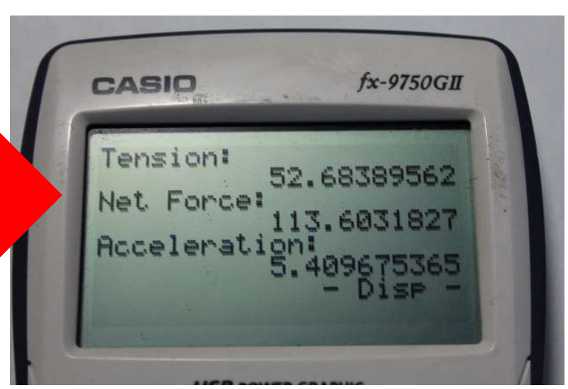












Yash Gharat
4B
Dr. Raj

Results

As with many programs, there were many positives and negatives based on the end-result. The Atwood machine solver, however, had mostly negatives. The programming language used, CASIO-Basic was not efficient at all. Learning it was very strange to the researcher because the syntax structure was unique. Overall, it was very inefficient to use this language. The most inefficient part of the program was the amount of hard-coding done. This was because there were a lot of equations that had to be standardized. If anything, other than these scenarios are used, the program would not be able to work. However, the program does fulfill its task and can solve simple Atwood machine problems.

The set-up of the program was created to be very user-friendly. There was a main menu which redirected you to the types of Atwood machines. These types included the “regular”, “modified” and “inclined” Atwood machines. Based on the selected type, the program redirected either to the input screen or to another menu screen which asked the user whether the problem needed the use of friction. For all scenarios, the proper values were asked for and stored into variables. The variable names were consistent throughout the entire program. The first mass was stored as A, the second mass was stored as B, the static coefficient of friction was stored as M, the kinetic coefficient of friction was stored as U, and the angle was stored as theta. The answer to the problem was then computed using these variables and derived equations. For all scenarios, the net force was found, followed by the acceleration, and then the tension force.

$$\sum F = \sum ma \text{ , Therefore } a = \frac{\sum F}{\sum m}$$

This was the derivation I used to solve for all scenarios. Each scenario had different forces involved. For example, the “regular” Atwood machine, used the difference of weight of the greater mass and the smaller mass. The smaller mass’s weight became the tension force. However, in the “modified”

Atwood machine, the weight of the first mass was negated by its normal force and the only force was the weight of the second mass. The net mass was just the sum of the two masses in all scenarios. This was accounted for in the program.

After giving the desired output, the program clears its screen and returns to the main menu to choose either another type of problem or exit. This was done using the "GetKey" command provided by the CASIO-Basic language. After the output is displayed, the program loops until the "EXE" key is pressed. At this point, it clears the screen and returns to main menu. I found this to be the most user-friendly feature of the entire program because it prevents the need from hitting run more than once.

Overall, this program was very successful because it fulfills its purpose completely. It may be hard to edit or add on to the program due to the logic of CASIO-Basic, but it will solve the most common problems with ease.

Conclusion

The researcher began this experiment pondering the question of whether he could construct a program that would automate the process of solving the tedious Atwood machine problems. That query can now be answered with a definite yes. Although the language is very limiting in its further applications, the objective was clearly met because the program can solve the three, most common types of Atwood machines seen on tests. This program could be greatly improved and expanded upon though.

The use of this program is very limiting it can only be used for three scenarios. Although the life of the average physics student could be greatly improved with this program, it could be even better if more scenarios were added and if there were diagrams to accompany each scenario. Also, it solves the problems for the user, but there is no explanation so that would be greatly helpful. Another issue I faced while making this was to derive equations which will always work for that scenario. However, this can't be the case because physics problems often change to test the understanding of the student. This is a real issue because the program would have to be robust so that it can handle these situations.

This experiment can be applied to real-life situations because students solve Atwood machine problems often in class. The researcher knows from experience as a student, that these problems not only take up a lot of time but can also become very messy if proper care is not taken to understand the problem. Many students tend to miss this problem simply because they mess up their calculation. For example, they may forget to add the second mass or they may use the wrong coefficient of friction.

This program eliminates that possibility by automating it. However, this program also has many restrictions to it. The greatest restriction is that it only works on Casio calculators. This is major because most students use Texas Instruments calculators. The researcher had to use CASIO-Basic because he only had access to a Casio calculator. The only way to fix this issue is to replicate the program using TI-

Basic. This may be a roundabout way of solving the issue but it is the only way. Luckily, the two languages are not far off from each other. They seemed to have been developed using similar ideas. Thus, it should not be too hard to translate.

Works Cited

- Atwood Machines." *Atwood Machines*. N.p., n.d. Web. 1 Dec. 2016. <http://www.aplusphysics.com/courses/honors/dynamics/Atwood.html>.
- "Atwood's Machine." *Atwood's Machine*. N.p., n.d. Web. 20 Nov. 2016. <http://hyperphysics.phy-astr.gsu.edu/hbase/atwd.html>.
- "Atwood's Machine." *Atwood's Machine*. N.p., n.d. Web. 24 Nov. 2016. <http://physics.kenyon.edu/EarlyApparatus/Mechanics/Atwoods_Machine/Atwoods_Machine.html>.
- *Chapter 8: Programming*. Place of Publication Not Identified: Book On Demand, 2013. Casio. Web. 23 Nov. 2016.
- Huhn_m. "Casio Basic Tutorial - Tutorials." *Universal Casio Forum*. N.p., 28 Nov. 2005. Web. 23 Nov. 2016. <http://community.casiocalc.org/topic/2448-casio-basic-tutorial/>.
- Review, Princeton. *Cracking the AP Physics 1 Exam*. New York: Penguin Random House, 2016. Print.